Building Ontology Networks: How to Obtain a Particular Ontology Network Life Cycle?

Mari Carmen Suárez-Figueroa
(Ontology Engineering Group, Departamento de Inteligencia Artificial
Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain
mcsuarez@fi.upm.es)

Asunción Gómez-Pérez
(Ontology Engineering Group, Departamento de Inteligencia Artificial
Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain
asun@fi.upm.es)

Abstract: To build an ontology, ontology developers should devise first a concrete plan for the ontology development, that is, they should establish the ontology life cycle. To do this, ontology developers should answer two key questions: a) which ontology life cycle model is the most appropriate for their ontology project? and b) which particular activities should be carried out in their ontology life cycle? In this paper we present a set of guidelines to help ontology developers and also naïve users answer such questions.

Keywords: Ontology engineering, Ontology development, ontology life cycle
Categories: I.2.4, M.2

1 Introduction

The methodological support for developing ontologies and ontology networks should include the *identification and definition of the development process*, *life cycle models* and *the life cycle*.

There are many different approaches for building ontologies. Thus, an analysis of methodologies was included in [Fernández-López, 02]; a series of existing methods and methodologies for developing ontologies from scratch have been reported in [Gómez-Pérez, 03]; a set of ontology learning methods for building ontologies was included in [Gómez-Pérez, 05]; and the experience of using wikis for gaining consensus on ontology modelling during the ontology development was reported in [Hepp, 07], among other approaches.

However, existing methodologies for building ontologies have some limitations with respect to the aforementioned issues. We analyzed such issues in three well known existing methodologies: METHONTOLOGY [Gómez-Pérez, 03], On-To-Knowledge [Staab, 01] and DILIGENT [Pinto, 04]).

With regard to the *identification and definition of the development process*, from the aforementioned methodologies, only METHONTOLOGY proposes explicitly a development process that identifies a set of activities performed during ontology development.

As for *life cycle models*, the three methodologies propose a unique life cycle model: METHONTOLOGY proposes an ontology life cycle model based on evolving prototypes; On-To-Knowledge proposes an incremental and cyclic ontology life cycle model based on evolving prototypes; and DILIGENT proposes an ontology life cycle model also based on evolving prototypes. However, it is well known in Software Engineering that there is no a unique life cycle model valid for all the developments.

Additionally, the literature lacks guidelines that help ontology developers to create a particular *ontology life cycle* based on a model.

To devise the concrete plan for the ontology development, two important questions have to be answered: 1) how do ontology developers decide which life cycle model is the most appropriate for their ontology? and 2) which particular activities should be carried out in their ontology life cycle? To respond to such questions, a collection of ontology life cycle models and some guidelines are presented in this paper. Such guidelines used an activity glossary (the so-called NeOn Glossary of Activities [Suárez-Figueroa, 08]) and the collection of models.

The rest of the paper is organized as follows: section 2 presents a collection of theoretical ontology life cycle models, section 3 explains the guidelines to obtain a particular ontology life cycle, and finally, section 4 includes some conclusions.

2 Ontology Network Life Cycle Models

An ***ontology network life cycle model*** is defined as the framework, selected by each organization, on which to map the activities identified and defined in the NeOn Glossary in order to produce the ontology network life cycle [Suárez-Figueroa, 07].

Within the Software Engineering field, it is acknowledged that there is not a unique life cycle model valid for all the software development projects and that each life cycle model is appropriate for a concrete project, depending on several features. For example, sometimes it is better a simple model (like waterfall [Royce, 70]), whereas other times it is most suitable a more complex one (like spiral [Boehm, 88]).

The same occurs in the Ontology Engineering field, where neither there is a unique model valid for all the ontology development projects, since each life cycle model is appropriate for a concrete development, depending on several features. Therefore, to propose a unique life cycle model for all the ontology network developments is not very realistic. Thus, taking into account the specific features of the ontology network development, a collection of theoretical ontology network life cycle models based on the models commonly used in Software Engineering has been created and proposed in [Suárez-Figueroa, 07]. These ontology network life cycle models vary from trivial and simple models to difficult and complex ones.

The proposed collection of models includes the following ones:

- *Waterfall life cycle model*. Its main characteristic is that it represents the stages of an ontology network as sequential phases. Thus, a concrete stage must be completed before the following stage begins.

  Because of the importance of knowledge resources reuse and reengineering and ontology merging, five significantly different versions of the waterfall ontology network life cycle model have been defined and proposed: (1) *five-phase waterfall*, (2) *six-phase waterfall* that extends the previous one with a new phase in which the reuse of already implemented ontological resources is considered,

(3) *six-phase + merging phase waterfall*, (4) *seven-phase waterfall* in which the six-phase model is taken as general basis and a new phase, the reengineering one, is included after the reuse phase, and (5) *seven-phase + merging phase*.

- *Incremental life cycle model*. Its main feature is that it divides the requirements in different parts and then develops each part in a different cycle. The idea is to incrementally "produce and deliver" the network of ontologies (full developed and functional), that is, the ontology network grows in layers (in a concentric way). Figure 1.a shows how an ontology network grows using this model (the striped parts in the figure mean the developed parts).
- *Iterative life cycle model*. Its main characteristic is that it divides all the requirements into small parts and develops the ontology network including requirements from all the parts. Figure 1.b shows how the ontology network is developed following this model (the striped parts mean the developed parts).

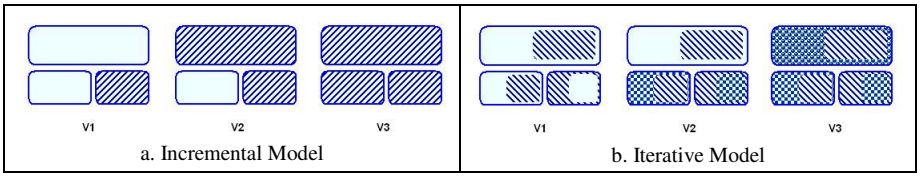


Figure 1: Schematic vision of an ontology network following (a) an incremental model and (b) an iterative model

- *Evolving prototyping life cycle model*. Its main feature is that it develops a partial product (in this case, partial ontology network) that meets the requirements best understood. The preliminary versions of the ontology network being developed (that is, the prototypes) permit the user to give feedback of unknown or unclear requirements.
- *Spiral life cycle model*. Its main feature is that it proposes a set of repetitive cycles based on waterfall and prototype models. In this model, taking into account the special characteristics of ontology networks, the space is divided into three sections: planning, risk analysis, and engineering. This division is based on the need to evaluate and assess all the outputs of all the ontology network stages, and not only after the engineering phase as it happens in software projects.

Relying on our own experience, we can briefly say that the waterfall ontology network life cycle model is the easiest model to understand, and that with this model it is also easy to schedule an ontology development. As for the incremental ontology network life cycle model, it permits to develop the ontology network having complete layers, following any type of waterfall model. Finally, the most sophisticated model is the spiral model that permits analyzing the different risks during the ontology network development.

3 Obtaining a Particular Ontology Network Life Cycle

The **ontology network life cycle** is defined as the project-specific sequence of activities created by mapping the activities identified in the NeOn Glossary of Activities onto a selected ontology network life cycle model [Suárez-Figueroa, 07]. The main objective of the *ontology network life cycle* is to determine when the activities identified should be carried out and through which stages the ontology network moves during its life.

Two key questions arise here: 1) how do ontology developers decide which ontology network life cycle model is the most appropriate for their ontology network? and 2) which particular activities should be carried out in their ontology network life cycle?

To help ontology developers to answer the above questions, we recommend the five steps presented in Figure 2. If they follow these steps, ontology developers will be able to answer both questions and to obtain the particular life cycle for their ontology network by mapping the selected ontology network life cycle model and the selected activities, and then ordering such activities.

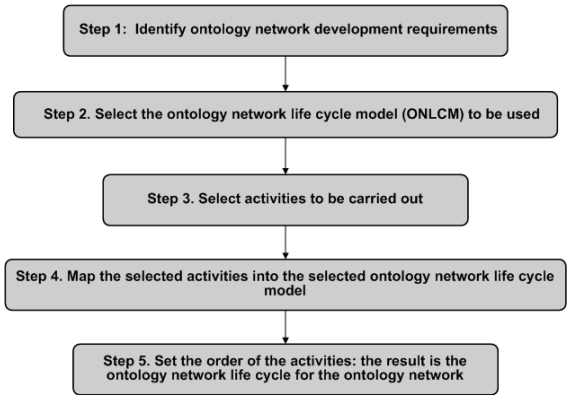


Figure 2: Steps for establishing the ontology network life cycle

Step 1: Identify ontology network development requirements. In this step, ontology developers identify the main needs of the ontology network development.

Step 2: Select the ontology network life cycle model (ONLCM) to be used. The main question here is: "which ontology network life cycle model should be chosen?". To carry out step 2, we propose the informal decision tree shown in Figure 3, which helps to select which ontology life cycle model is the most appropriate for the ontology network being built.

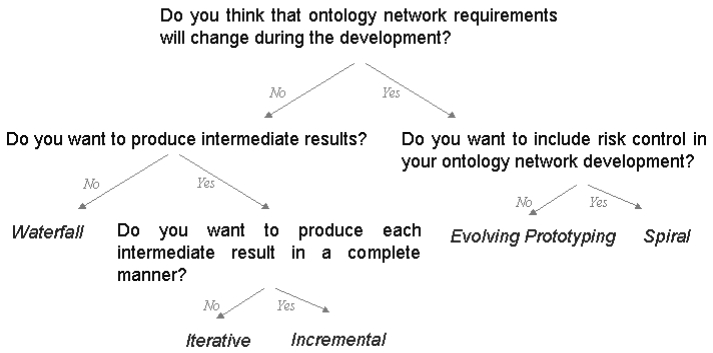


Figure 3: Decision tree for selecting the ontology network life cycle model

Step 3: Select activities to be carried out. Activities potentially involved in the ontology network development process are defined in the NeOn Glossary of Activities[1] [Suárez-Figueroa, 08]. In order to facilitate ontology developers the selection of activities from the NeOn Glossary for a concrete development, we have distinguished between required and if applicable activities.

- *Required or Mandatory activities* refer to those activities that should be carried out when developing networks of ontologies. The activities identified as "required" can be considered as core for the ontology development.
- *If Applicable or Optional activities* refer to those activities that can be carried out or not, depending on the case, when developing ontology networks.

To group the activities of the NeOn Glossary into one of the two previous categories, we made an open call and invited ontology developers participating in international projects (NeOn, KWeb, X-Media, etc.) and working in universities and companies (DERI group, OEG group, iSOCO, etc.) to participate in an on-line survey[2]. This survey began on July 27th 2007 and the results were collected on August 21st 2007. It was answered by thirty five people.

The table of 'Required-If Applicable' activities, which is shown in Table 1, has been built considering the results of this survey and our own experience on developing ontologies. The table includes all the activities identified and defined in the NeOn Glossary.

Required activities plus all others applicable to the ontology network development should be selected to be carried out during the ontology network life cycle. The result of step 3 is the table of selected activities. In this step, we propose to distinguish between two distinct kinds of ontology developers:

- *Experienced Ontology Developers*. We assume that, drawing on their own experience, ontology developers are able to select the activities to be carried out during the ontology network life cycle from the "Required-If Applicable" table. Activities identified as "required" in the "Required-If Applicable" table are

[1] http://www.neon-project.org/web-content/images/Publications/neonglossaryofactivities.pdf.
[2] http://droz.dia.fi.upm.es/survey/index.jsp

selected automatically. Ontology developers should only select those "if applicable" activities they need for their ontology network development.

- *Naïve Ontology Developers*. For those "if applicable" activities, we propose a list of "yes/no" natural language questions (some examples are shown in Table 2) to be answered by naïve ontology developers. If the response of a concrete question is positive, then the corresponding activity is selected; otherwise, the activity is not selected. As in the previous case, activities identified as "required" in the "Required-If Applicable" table are selected automatically.

Required		***If Applicable***	
O. Annotation	O. Configuration Management	O. Aligning	O. Forward Engineering
O. Assessment	Control	O. Customization	Ontology Learning
O. Comparison	O. Diagnosis	O. Enrichment	O. Localization
O. Conceptualization	O. Documentation	O. Extension	O. Matching
O. Elicitation	O. Feasibility Study	O. Merging	O. Partitioning
O. Environment Study	O. Formalization	O. Modification	O. Population
O. Evaluation	O. Implementation	O. Modularization	O. Pruning
O. Evolution	O. Integration	O. Module Extraction	Non Ontological Resource Reengineering
Knowledge Acquisition for Ontologies	Scheduling	O. Reengineering	O. Specialization
O. Quality Assurance	O. Search	O. Restructuring	O. Summarization
O. Repair	O. Selection	Non Ontological Resource Reuse	O. Translation
O. Reuse	O. Specification	O. Reverse Engineering	O. Update
O. Upgrade	O. Verification		
O. Validation	O. Versioning		

Table 1: Required-If Applicable Activities

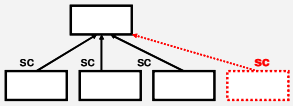
Activity	Natural Language Questions
Ontology Customization	Do you wish to modify the ontology network to meet specific user's needs?
Ontology Extension	Do you wish to stretch, widen, broaden or expand your current ontology network by adding new concepts "in a horizontal way/direction" to widen its sphere of action? 
Ontology Localization	Do you wish to have your ontology network in different natural languages, as for example, in English, Spanish and/or French?
Ontology Reengineering	Do you wish to take an existing and implemented ontology to enhance it and implement it again?
Non Ontological Resource Reuse	Do you intend to use non ontological resources (such as a controlled vocabularies or data bases) in the development of your ontology?

Table 2: Examples of Proposed "Yes/No" Natural Language Questions

Step 4: Map the selected activities into the selected ontology network life cycle model. To carry out this mapping, ontology developers should match the selected activity outputs against the requirements of each phase or stage in the selected ONLCM. This step provides an activity map or matrix for the ontology network development.

Step 5: Set the order of the activities: the result is the ontology network life cycle for the ontology network. After obtaining the activity map or matrix, ontology developers should order the activities of this matrix, thus obtaining the ontology network life cycle. The order in which the activities will be performed are determined by three major factors:

- The selected ONLCM dictate an initial ordering of activities.
- Schedule constraints may require the overlapping of activities in the ONLCM and may thus impact the ordering.
- Selection and ordering of activities might be impacted by the entry and exit criteria of associated activities. The availability of output information from one activity could affect the start of another activity.

The guidelines proposed in this paper are being used and thus evaluated in the development of the ontologies in two use cases within the NeOn project [Suárez-Figueroa, 07]: invoice management and semantic nomenclature, both belonging to the pharmaceutical domain.

4 Conclusions

The main contribution of our paper is the set of guidelines we have created to help ontology developers obtain the concrete life cycle of an ontology network.

Our guidelines for obtaining the concrete life cycle for an ontology network are mainly created to help ontology developers to make these two decisions: (1) selecting the ontology network life cycle model that is the most appropriate for a concrete case and (2) selecting which activities, from the NeOn Glossary of Activities, should be carried out.

Thus, for the first decision, we propose some guidelines involving the collection of ontology network life cycle models presented in this paper. Such models are based on the models defined in the Software Engineering field and take into account the specific features of the ontology network development.

For the second decision, we suggest some guidelines that use the NeOn Glossary of Activities, which identifies and defines the activities potentially involved in the ontology network development. The activities in the NeOn Glossary have been divided into activities required for ontology network development and those that could or could not be applicable, depending on the concrete case, and consequently non-essential or dispensable. The proposed guidelines are founded on natural language questions for helping naïve users to select the activities they have to perform.

Acknowledgements

This work has been supported by the NeOn project (IST-2005-027595). We are very grateful to Elena Montiel-Ponsoda for her help with the natural language questions.

References

[Boehm, 88] Boehm, B. W.: *A spiral model of software development and enhancement*. ACM SIGSOFT Software Engineering Notes. Vol. 11, no. 4, pp. 14-24, August 1986; reprinted in Computer, vol. 21. no. 5, pp. 61-72, May 1988.

[Fernández-López] Fernández-López, M., Gómez-Pérez, A.: *Overview and Analysis of Methodologies for Building Ontologies*. Knowledge Engineering Review (KER). Vol. 17, Nº 2, pp 129-156. 2002.

[Gómez-Pérez, 05] Gómez-Pérez, A., Manzano-Macho, D.: *An overview of methods and tools for ontology learning from texts*. Knowledge Engineering Review, 19:187–212, 2005.

[Gómez-Pérez, 03] Gómez-Pérez, A., Fernández-López, M., Corcho, O.: *Ontological Engineering*. November 2003. Springer Verlag. Advanced Information and Knowledge Processing series. ISBN 1-85233-551-3.

[Haase, 06] Haase, P., Rudolph, S., Wang, Y., Brockmans, S., Palma, R., Euzenat, J., d'Aquin, M.: *NeOn Deliverable D1.1.1 Networked Ontology Model*. November 2006. Available at: http://www.neon-project.org/.

[Hepp, 07] Hepp, M., Siorpaes, K., Bachlechner, D.: *Harvesting Wiki Consensus: Using Wikipedia Entries as Vocabulary for Knowledge Management*. IEEE Internet Computing 11(5): 54-65. 2007.

[Pinto, 04] Pinto, H. S., Tempich, C., Staab, S.: *DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies*. In Ramón López de Mantaras and Lorenza Saitta, Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), August 22nd - 27th, pp. 393--397. IOS Press, Valencia, Spain, August 2004. ISBN: 1-58603-452-9. ISSN: 0922-6389.

[Royce, 70] Royce, W. W.: *Managing the development of large software systems: concepts and techniques*. In Proceedings Western Electronic Show and Convention (WESCON), Los Angeles. August 25th-28th 1970.

[Staab, 01] Staab, S., Schnurr, H.P., Studer, R., Sure, Y.: *Knowledge Processes and Ontologies*. IEEE Intelligent Systems 16(1):26–34. (2001).

[Suárez-Figueroa, 07] Suárez-Figueroa, M. C., Aguado de Cea, G., Buil, C., Caracciolo, C., Dzbor, M., Gómez-Pérez, A., Herrero, G., Lewen, H., Montiel-Ponsoda, E., Presutti, V.: *NeOn Deliverable D5.3.1 NeOn Development Process and Ontology Life Cycle*. August 2007. Available at: http://www.neon-project.org/.

[Suárez-Figueroa, 08] Suárez-Figueroa, M. C., Gómez-Pérez, A.: *Towards a Glossary of Activities in the Ontology Engineering Field*. 6th Language Resources and Evaluation Conference (LREC 2008). Marrakech (Morocco). May 2008.